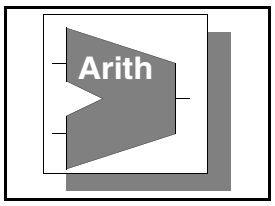
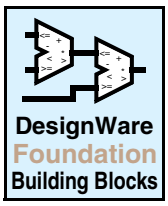



# DW01_cmp2

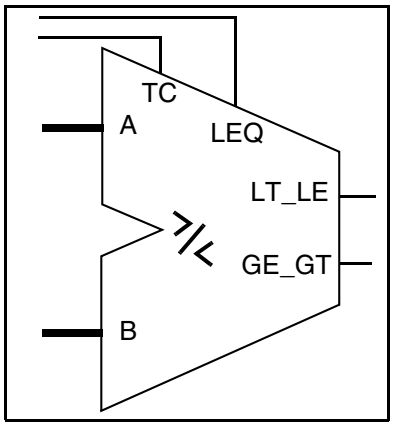
## 2-Function Comparator

Version, STAR, and myDesignWare Subscriptions: IP Directory

## Features and Benefits

- Parameterized word length
- Unsigned and signed (two's-complement) data operation

Revision History




## Description

DW01_cmp2 is a two-input comparator. DW01_cmp2 compares two signed or unsigned numbers A and B and produces two output conditions LT_LE and GE_GT as results.

**Table 1-1 Pin Description**

| Pin Name | Width | Direction | Function |
|---|---|---|---|
| A | *width* bits | Input | Input data |
| B | *width* bits | Input | Input data |
| LEQ | 1 bit | Input | Output condition control |
| TC | 1 bit | Input | Two's complement control<br>■ 0 = Unsigned<br>■ 1 = Signed |
| LT_LE | 1 bit | Output | Less-than/less-than-or-equal output condition |
| GE_GT | 1 bit | Output | Greater-than-or-equal/greater-than output condition |

**Table 1-2 Parameter Description**

| Parameter | Values | Description |
|---|---|---|
| width | ≥ 1 | Word length of A and B |

**Table 1-3 Synthesis Implementations**[a]

| Implementation Name | Function | License Feature Required |
|---|---|---|
| rpl | Ripple-carry synthesis model | none |
| pparch | Delay-optimized flexible parallel-prefix | DesignWare |
| apparch | Area-optimized flexible architecture that can be optimized for area, for speed, or for area, speed | DesignWare |

a. During synthesis, Design Compiler will select the appropriate architecture for your constraints. However, you may force Design Compiler to use any architectures described in this table. For more, see *DesignWare Building Block IP User Guide*.

**Table 1-4 Simulation Models**

| Model | Function |
|---|---|
| DW01.DW01_CMP2_CFG_SIM | Design unit name for VHDL simulation |
| dw/dw01/src/DW01_cmp2_sim.vhd | VHDL simulation model source code |
| dw/sim_ver/DW01_cmp2.v | Verilog simulation model source code |

**Table 1-5 Functional Description**

| LEQ | Condition | LT_LE | GE_GT |
|---|---|---|---|
| 1 | A ≤ B | 1 | 0 |
| 1 | A > B | 0 | 1 |
| 0 | A < B | 1 | 0 |
| 0 | A ≥ B | 0 | 1 |

The input signal LEQ determines whether the two output conditions are LT (less-than) and GE (greater-than-or-equal) (LEQ = 0) or LE (less-than-or-equal) and GT (greater-than) (LEQ = 1). The input TC determines whether the two inputs are compared as unsigned (TC = 0) or signed two's complement (TC = 1) numbers.

## Related Topics

- Math – Arithmetic Overview
- DesignWare Building Block IP User Guide

## HDL Usage Through Operator Inferencing - VHDL

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity DW01_cmp2_oper is
  generic(wordlength: integer:=8);
  port(in1, in2    : in STD_LOGIC_VECTOR(wordlength-1 downto 0);
       instruction : in STD_LOGIC;
       comparison  : out boolean);
end DW01_cmp2_oper;

architecture oper of DW01_cmp2_oper is
  signal in1_signed, in2_signed: SIGNED(wordlength-1 downto 0);
begin
  in1_signed <= SIGNED(in1);
  in2_signed <= SIGNED(in2);
  -- infer the non-equality comparison operators
  process (in1_signed, in2_signed, instruction)
  begin
    if (instruction = '0') then
      comparison <= in1_signed > in2_signed;
    else
      comparison <= in1_signed >= in2_signed;
    end if;
  end process;
end oper;
```

## HDL Usage Through Operator Inferencing - Verilog

```
module DW01_cmp2_oper(in1, in2, instruction, comparison);
  parameter wordlength = 8;

  input [wordlength-1:0] in1, in2;
  input instruction;
  output comparison;
  reg comparison;

  always @ (in1 or in2 or instruction)
  begin
    if (instruction == 0)
      comparison = (in1 > in2);
    else
      comparison = (in1 >= in2);
  end
endmodule
```

## HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW01_cmp2_inst is
  generic ( inst_width : NATURAL := 8 );
  port ( inst_A : in std_logic_vector(inst_width-1 downto 0);
         inst_B : in std_logic_vector(inst_width-1 downto 0);
         inst_LEQ : in std_logic;
         inst_TC : in std_logic;
         LT_LE_inst : out std_logic;
         GE_GT_inst : out std_logic );
end DW01_cmp2_inst;

architecture inst of DW01_cmp2_inst is
begin

  -- Instance of DW01_cmp2
  U1 : DW01_cmp2
    generic map ( width => inst_width )
    port map ( A => inst_A, B => inst_B, LEQ => inst_LEQ,
               TC => inst_TC, LT_LE => LT_LE_inst, GE_GT => GE_GT_inst );
end inst;

-- pragma translate_off
configuration DW01_cmp2_inst_cfg_inst of DW01_cmp2_inst is
  for inst
  end for; -- inst
end DW01_cmp2_inst_cfg_inst;
-- pragma translate_on
```

## HDL Usage Through Component Instantiation - Verilog

```
module DW01_cmp2_inst( inst_A, inst_B, inst_LEQ, inst_TC,
                       LT_LE_inst, GE_GT_inst );

  parameter width = 8;

  input [width-1 : 0] inst_A;
  input [width-1 : 0] inst_B;
  input inst_LEQ;
  input inst_TC;
  output LT_LE_inst;
  output GE_GT_inst;

  // Instance of DW01_cmp2
  DW01_cmp2 #(width)
    U1 ( .A(inst_A), .B(inst_B), .LEQ(inst_LEQ), .TC(inst_TC),
         .LT_LE(LT_LE_inst), .GE_GT(GE_GT_inst) );

endmodule
```

## Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

| Date | Release | Updates |
|---|---|---|
| March 2019 | DWBB_201903.0 | ■ Removed Table 1-4, “Obsolete Synthesis Implementations” |
| January 2019 | DWBB_201806.5 | ■ Updated example in “HDL Usage Through Component Instantiation - VHDL” on page 5<br>■ Added this Revision History table and the document links on this page |

# Copyright Notice and Proprietary Information



### Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

### Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

### Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

### Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

### Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com